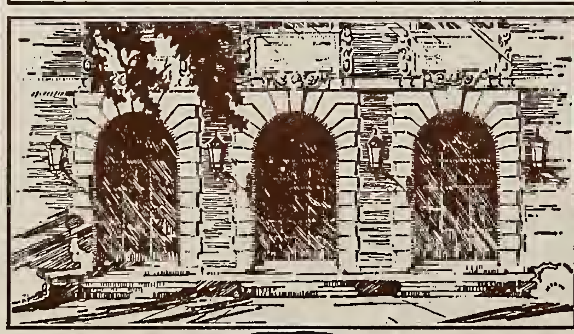

UIUCDCS-R-72-542

THE STATUS OF INVESTIGATIONS INTO THE USE OF CONTINUED FRACTIONS FOR COMPUTER HARDWARE

by

James E. Robertson*
and Kishor Trivedi*

August, 1972

DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN · URBANA, ILLINOIS

UIUCDCS-R-72-542

THE STATUS OF INVESTIGATIONS INTO THE USE OF CONTINUED FRACTIONS FOR COMPUTER HARDWARE

by

James E. Robertson*
and Kishor Trivedi*

August, 1972

The research on which this paper is based was supported in part by the National Science Foundation under Contract GJ813, by the Applied Mathematics Department of Brookhaven National Laboratory, and by the Department of Computer Science of the University of Illinois.

*Department of Computer Science, University of Illinois, Urbana, Illinois.

Abstract

The purpose of this paper is to demonstrate that representations of numbers other than positional notation may lead to practical hardware realizations for the digital calculation of classes of algorithms. It is the authors' opinion that practicality of the use of continued products has been demonstrated. This paper describes current research in the use of continued fractions. Although practicality has not been demonstrated, theoretical results are promising, and the results thus far are presented as a case study of the difficulties which arise when use of a new representation is attempted.

There appear to be three fundamental requirements for a proposed representation to be useful. Existence of sets of digital values for coefficients (e.g., for partial denominators and partial numerators for continued fractions) and a concomitant simple procedure for conversion to positional notation are necessary. Secondly, a compatible set of algorithms must be found, with the set of sufficient scope to justify its use. Thirdly, simple rules for selection of the coefficients must be found.

For continued fractions, the first requirement necessitates the use of simple binary coefficients (e.g., 1/2, 1), which primarily distinguishes the approach taken here from results of the past several centuries of research. For the second requirement, emphasis is given to the first algorithm to be found, namely, solution of a quadratic equation. Finding a set of algorithms of sufficiently wide scope seems to be the major problem, and at this time, only a few techniques which may lead to new algorithms can be described. An example of selection rules for solution of the quadratic is presented in detail.

1. History and Motivation

This paper is essentially a report on research in progress. The fundamental observation is that, currently, virtually all digital hardware calculations are based on the use of positional notation; equivalently, on weighted sums of series. Other representations of numbers exist; the concern here will be with continued products and continued fractions.

The use of positional notation has been limited to addition, subtraction, multiplication, division, and, to a lesser extent, square and higher roots. It has been shown [1] that use of continued products extends the list of implementable algorithms to the logarithm, the exponential, the trigonometric and inverse trigonometric functions, as well as multiply, divide, and square root. Both time of execution and cost of hardware are reasonable with current technology; in comparison with a conventional arithmetic unit, factors of 2 to 3 for both time and cost are typical. A small read-only memory fast enough to match accumulator rates is also needed. The investigation of the use of continued products was originally limited to the binary case. Higher radix techniques appear promising, and are being investigated [2]. Otherwise, emphasis will be given here to investigations into the use of continued fractions. Results to date are theoretically promising, but not yet practical in the sense of hardware implementation.

There appear to be three fundamental requirements for a proposed representation of numbers to be useful for implementation in hardware.

These are:

1) Conversion to conventional series form (positional notation) must be both possible and simple. Implicit here is the requirement that the set of possible results spans continuously (in the limit of infinite precision) some permissible range of values. For floating point arithmetic, it seems sufficient to require that the ratio of the upper limit to the lower limit of the range be at least two.

2) The set of algorithms should include algorithms which are easily soluble for the representation of numbers employed. Compatability among algorithms, in the sense of hardware sharing, is also a desirable goal.

3) Since most algorithms, other than multiplication, appear to require trial and error procedures in the absence of redundancy, it must be possible to devise techniques such that the selection of each of the successive coefficients is practical (cf., quotient digit selection in division).

It should be pointed out that the use of the coefficients of a representation is ephemeral, since conversion to positional notation occurs in parallel with the successive steps of the algorithm. For example, for a continued product,

$$\prod_{i=1}^{k} (1+2^{-i}\epsilon_i) = (1+2^{-k}\epsilon_k)\prod_{i=1}^{k-1}(1+2^{-i}\epsilon_i).$$

At any one step the (k-1)st continued product has been determined, the coefficient ϵ_k is determined by the selection rules appropriate to the algorithm, and the kth value of the continued product (in positional notation) is calculated by adding the (k-1)st value to a shifted version

of itself. The entire set of coefficients ϵ_i ($i = 1, 2, \ldots, m$) is never simultaneously available.

It is difficult to generalize about the procedures necessary to determine whether or not a proposed representation satisfies the requirements previously discussed. For continued products and continued fractions, determination of the set of coefficient values and the associated conversion procedure has been relatively simple. The identification of suitable algorithms appears to be by far the most difficult requirement to satisfy. In retrospect, for continued products, the observation that the logarithm of a continued product is the sum of the logarithms of the individual terms leads to the identification of the logarithm and its inverse, the exponential, as suitable algorithms. Similarly, the properties of the complex exponential indicate that the trigonometric functions and their inverses are identifiable as algorithms for continued product representations. No such general observation is as yet apparent to the authors' for continued fractions. Formulating selection rules appears to be very much a function of the individual algorithm.

2. Examples: Division Algorithms

For illustrative purposes, algorithms for division are developed for positional notation, for continued products, and for continued fractions. In each case, the initial assumption is that

$$\frac{N}{D} - X \to 0$$

or some variant thereof, where N is the dividend, D is the divisor, and X is the quotient. The same procedures are then used for developing the algorithms, except for the representation of X. For positional notation the division algorithm in common use is developed. For continued products, a new algorithm with many useful properties is found. The continued fraction algorithm is obviously an exercise in futility, since the conversion procedure requires of itself a division as its terminal step.

2.1 Positional Notation

We define the remainder at the ith step by

$$N - DX_i = Y_i, \text{ and also } N - DX_{i-1} = Y_{i-1}$$

For positional notation

$$X_i = X_{i-1} + 2^{-i}x_i = \sum_{j=1}^{i} 2^{-j}x_j$$

$$Y_i = N - DX_i = N - DX_{i-1} - 2^{-i}Dx_i = Y_{i-1} - 2^{-i}Dx_i$$

Since the allowed range of Y_i decreases by a factor of two at each step, it is convenient to define a shifted remainder r_i

$$r_i = 2^iY_i, \text{ and also } r_{i-1} = 2^{i-1}Y_{i-1}$$

$$r_i = 2^iY_{i-1} - Dx_i = 2r_{i-1} - Dx_i \qquad 2.1.1$$

Equation 2.1.1 is the familiar recursion for most binary division procedures in common use. The initial remainder $Y_o = r_o = N$, the dividend, and $X_o = 0$.

2.2 Continued Products

As in the previous example, the remainder is

$$N - DX_i = Y_i, \text{ and } N - DX_{i-1} = Y_{i-1}$$

For a continued product

$$X_i = X_{i-1}(1+2^{-i}x_i) = \prod_{j=1}^{i} (1+2^{-j}x_j) \qquad 2.2.1$$

$$Y_1 = N - DX_i = N - DX_{i-1} - 2^{-i}DX_{i-1}x_i$$

$$= Y_{i-1} - 2^{-i}(N-Y_{i-1})x_i$$

$$= Y_{i-1}(1+2^{-i}x_i) - 2^{-i}Nx_i$$

It is again convenient to define a shifted remainder r_i.

$$r_i = 2^iY_i, \text{ and } r_{i-1} = 2^{i-1}Y_{i-1}$$

$$r_i = 2^iY_{i-1}(1+2^{-i}x_i) - Nx_i$$

$$= 2r_{i-1}(1+2^{-i}x_i) - Nx_i \qquad 2.2.2$$

The conversion procedure is

$$X_i = X_{i-1} + 2^{-i}x_iX_{i-1}, \text{ with } X_o = 1$$

and $x_i = \bar{1}, 0, 1$, or $x_i = 0, 1$. The initial remainder is $Y_o = r_o = N - D$.

An alternative which simplifies the selection procedure is to let $N = 1$ in equation 2.2.2, and compensate by letting $X_o = N$ as the initial condition for the conversion procedure of equation 2.2.1.

2.3 Continued Fractions

For continued fractions, let $X_i = \frac{P_i}{Q_i}$, and define the remainder Y_i as

$$Y_i = NQ_i - DP_i$$

The conversion procedure is given by the recursions

$$\begin{aligned} P_i &= q_i P_{i-1} + p_i P_{i-2} \qquad P_o = 0 \qquad P_1 = p_1 \\ Q_i &= q_i Q_{i-1} + p_i Q_{i-2} \qquad Q_o = 1 \qquad Q_1 = q_1 \end{aligned} \tag{2.3.1}$$

which must be followed by a terminal division (i=m) as indicated by $X_m = \frac{P_m}{Q_m}$. Otherwise, the conversion consists of additions and shifts if p_i and q_i are simple binary coefficients; e.g., 1/4, 1/2, 1, and 2.

We note that

$$Y_{i-2} = NQ_{i-2} - DP_{i-2}$$

$$Y_{i-1} = NQ_{i-1} - DP_{i-1}$$

therefore

$$\begin{aligned} Y_i &= N(q_i Q_{i-1} + p_i Q_{i-2}) \\ &\quad - D(q_i P_{i-1} + p_i P_{i-2}) \end{aligned}$$

$$Y_i = q_i Y_{i-1} + p_i Y_{i-2} \tag{2.3.2}$$

Equation 2.3.2 is derived here for illustrative purposes only. Due to the obvious impracticality of the process, neither the rate of convergence (i.e., decrease in range of Y_i) nor the selection procedure (i.e., method of choosing q_i and p_i) have been studied.

3. The First Quadratic

Consider the finite continued fraction with k partial numerators p_i and k partial denominators q_i $(i = 1, 2, \ldots, k)$, whose value is P_k/Q_k, i.e.,

$$\frac{P_k}{Q_k} = \cfrac{p_1}{q_1 + \cfrac{p_2}{q_2 + \cfrac{p_3}{q_3 + \ddots + \cfrac{p_k}{q_k}}}}$$

P_k and Q_k are determined from the recursions:

$$P_i = q_i P_{i-1} + p_i P_{i-2} \qquad i = 2, 3, \ldots, k$$

$$Q_i = q_i Q_{i-1} + p_i Q_{i-2} \qquad P_0 = 0 \quad P_1 = p_1 \quad Q_0 = 1 \quad Q_1 = q_1$$

It is clear that P_k and Q_k can be separately and simultaneously determined in two binary arithmetic units in k-1 addition times if the p_i and q_i are chosen to be simple in the binary sense. It is convenient to make the choice $p_i = 1$ for all i; it can be shown (Section 6) that other values of p_i are admissable. The digit set for q_i is initially assumed to be two-valued, and after some investigation it was found that choice of the digit set $q_i = \{1/2, 1\}$ yields continued fractions whose values $\frac{P_k}{Q_k}$ are continuous in the limit over the interval as defined by the following equation:

$$1/2 \leq \lim_{k \to \infty} \frac{P_k}{Q_k} \leq 1$$

These properties indicate that a suitable continued fraction representation exists, such that conversion to conventional binary can be achieved by repetitive use of two binary adders in parallel, followed by a division to determine the quotient P_k/Q_k.

Determination of an algorithm and the appropriate corresponding computational procedure is much more difficult. The particular algorithm chosen for investigation was the solution of the limited class of quadratics

$$x^2 + b_k x - c_k = (x-u)(x+v) = 0 \qquad 3.1$$

such that $1/2 \leq u \leq 1$. The problem, specifically, is, given b_k and c_k, find u (and hence $v = b_k + u$). This algorithm was selected because of the following property of infinite periodic continued fractions, of period k. If the sum of the first k-1 terms is P_{k-1}/Q_{k-1} and the sum of the first k terms is P_k/Q_k, then the quadratic coefficients b_k and c_k are $b_k = (Q_k - P_{k-1})/Q_{k-1}$ and $c_k = P_k/Q_{k-1}$. The value of the infinite periodic continued fraction is then u, the positive root of the quadratic. The problem is then resolved specifically to the following one. Given $(Q_k - P_{k-1})/Q_{k-1}$ and P_k/Q_{k-1} (Note that k is unknown.), find the sequence of partial denominators q_i $(i = 1, 2, \ldots, k)$.

The first two approaches to a computational procedure were similar. Given the limited information about the values of continued fractions of order k-1 and order k implicit in b_k and c_k, determine the sequence of partial denominators

a) $q_{k+1} = q_1, q_{k+2} = q_2, \ldots, q_{2k} = q_k$, or

b) $q_k, q_{k-1}, \ldots, q_1$

After extensive investigation, it appears possible to prove that either of these approaches requires knowledge of the solution in order to determine the q_i, hence these approaches were abandoned.

The third and fourth approaches were based on the observation that the value u of the infinite periodic continued fraction of period k with $p_i = 1$, $q_i = \{1/2, 1\}$, $(i = 1, 2, \ldots, k)$ is also the value u of the infinite periodic continued fraction of period one with each $p_i = c_k$ and each $q_i = b_k$ $(i = 1, 2, \ldots, \infty)$. That is,

$$u = \frac{c_k}{b_k + u}.$$

Approaches three and four may therefore be considered as methods of conversion from one form of infinite continued fraction to that form which is easily converted to a conventional binary representation.

Computational approach number three, in successive steps, generates q_1, q_2, etc., such that

$$u = \cfrac{c_k}{b_k + \cfrac{c_k}{b_k + \cfrac{c_k}{b_k + \ddots}}} = \cfrac{1}{q_1 + \cfrac{\gamma_{k-1}}{\beta_{k-1} + \cfrac{\gamma_{k-1}}{\beta_{k-1} + \ddots}}} = \cfrac{1}{q_1 + \cfrac{1}{q_2 + \cfrac{\gamma_{k-2}}{\beta_{k-2} + \cfrac{\gamma_{k-2}}{\beta_{k-2} + \ddots}}}},$$

For this approach, the recursion relations are

$$\beta_{k-n} = 2q_n - \frac{\beta_{k-n+1}}{\gamma_{k-n+1}} \text{ and } \gamma_{k-n} = \frac{1 + q_n\beta_{k-n+1}}{\gamma_{k-n+1}} - q_n^2$$

for $n = 1, 2, \ldots,$ with $\beta_k = b_k$ and $\gamma_k = c_k$ for the first step. Clearly, this procedure requires two divisions as well as other operations at each step, and is unsuitable for mechanization.

The fourth approach, in successive steps, generates partial quotients q_1, q_2, etc., by increasing the periodicity of periodic continued fractions, as follows:

$$u = \cfrac{c_k}{b_k + \cfrac{c_k}{b_k + \cfrac{c_k}{b_k + \ddots}}} = \cfrac{1}{q_1 + \cfrac{c_{k-1}}{b_{k-1} + \cfrac{1}{q_1 + \cfrac{c_{k-1}}{b_{k-1} + \ddots}}}} = \cfrac{1}{q_1 + \cfrac{1}{q_2 + \cfrac{c_{k-2}}{b_{k-2} + \ddots}}}$$

After a considerable amount of algebra, the recursion relations can be shown to be:

$$b_{k-n} = q_n\, c_{k-n+1} - q_{n-1}\, c_{k-n+2} + b_{k-n+2}$$

$$c_{k-n} = -q_n\, b_{k-n} + q_n\, b_{k-n+1} + c_{k-n+2} \qquad 3.2$$

It should be noted that the relative simplicity of these recursions is dependent on the fact that $P_n Q_{n-1} - P_{n-1} Q_n = (-1)^{n-1}$, which requires that the partial numerators p_i be 1. For n = 1, the recursions require that $b_{k+1} = 0$, $c_{k+1} = 1$, and $q_0 = 0$.

4. Extension of the Range and Domain of Quadratic Solutions

In the previous section, the generality of the solution of the quadratic of equation 3.1 is limited by the requirement that the root u is representable. For the choice $q_i \epsilon \{1/2, 1\}$ and $p_i = 1$, the range of u is $1/2 \leq u \leq 1$. Replacing x in equation 3.1 by $u_{min} = 1/2$ and $u_{max} = 1$, the solutions are limited to the triangular wedge in the c_k, b_k plane

$$1/2\, b_k + 1/4 \leq c_k \leq b_k + 1 \qquad (4.1)$$

It will be shown in section 5 that selection procedures impose the further requirement $b_k \geq 0$. (4.2) The purpose of this section is to show that any point in the upper half of the c_k, b_k plane (i.e., $c_k > 0$) can be mapped onto a point in the region defined by conditions 4.1 and 4.2.

At this point, it is convenient to delineate four areas in the c_k, b_k plane and relate each area to properties of the root magnitudes u and v.

1) $c_k < -\frac{b_k^2}{4}$. Both roots are imaginary.
2) $-\frac{b_k^2}{4} \leq c_k < 0$. Both roots are real and of the same sign.
3) $c_k \geq 0, b_k < 0$ (second quadrant). The roots are real and of opposite sign, with $u > v$.
4) $c_k \geq 0, b_k \geq 0$ (first quadrant). The roots are real and of opposite sign, with $v \geq u$.

It is first shown that any point in the first quadrant of the c_k, b_k plane may be scaled to lie within a triangular wedge such that $1/2 \leq u \leq 1$. Since $v = u + b_k$ and $c_k = uv$, it follows that $c_k = ub_k + u^2$, and the range $1/2 \leq u \leq 1$ is equivalent to

$$1/2\, b_k + 1/4 \leq c_k \leq b_k + 1 \qquad (4.1)$$

Multiplying equations 3.1 and 4.1 by 2^{2j} (j an integer) yields

$$(2^j x)^2 + (2^j b_k)(2^j x) - 2^{2j} c_k = 0 \tag{4.3}$$

$$2^{j-1}\,(2^j b_k) + 2^{2(j-1)} \leq 2^{2j} c_k \leq 2^j\,(2^j b_k) + 2^{2j} \tag{4.4}$$

Let $2^j x = x'$, $2^j b_k = b_k'$, and $2^{2j} c_k = c_k'$. Then

$$(x')^2 + b_k'\,x' - c_k' = 0 \tag{4.5}$$

$$2^{j-1}\,b_k' + 2^{2(j-1)} \leq c_k' \leq 2^j\,b_k' + 2^{2j} \tag{4.6}$$

Given c_k' and b_k', the scaling procedure is then

a) Determine the value of j, such that equation 4.6 is satisfied.

b) Multiply c_k' and b_k' by 2^{-2j} and 2^{-j}, respectively, to obtain c_k and b_k, which satisfy equation 4.1.

c) When the root u is determined, find the positive root u' of equation 4.5, by scaling u in accordance with $u' = 2^j u$.

Note that the scaling procedure reduces to that normally employed for square roots in floating point computers, when $b_k = 0$.

For any point c_k', b_k' in the first quadrant, an integer value of j can be found such that equation 4.6 is satisfied. It is therefore sufficient, for the first quadrant, to solve equation 3.1 subject to the constraints of equation 4.1, with $b_k \geq 0$.

For the second quadrant, with $b_k < 0$ it is sufficient to replace $b_k = v - u$ by $b_k'' = -b_k = u - v$. Equation 3.1 becomes

$$x^2 + b_k''\,x - c_k = (x-v)(x+u) = 0. \tag{4.7}$$

Solution of (4.7) yields the magnitude v of the negative root. The value of u is then $u = b_k'' + v$.

The solution for the case of two imaginary roots has not been considered. Attempts to find a method of solution for two real roots of the same sign have thus far been unsuccessful. The preceding observations, however, indicate that a continued fraction solution of the quadratic can be found if $c_k > 0$; i.e., if the two roots are real and are of opposite sign.

5. Selection Procedures for the First Quadratic

This section develops a selection procedure for p_i and q_i of the algorithm of equations 3.2 for solving quadratics using continued fractions. We decide to have $p_i = 1$ for all i. Thus the problem reduces to the selection of q_i.

First, we must choose the set from which to pick q_i; we call this a digit set of q_i. We put five requirements on this digit set.

a) All elements must be of the form 2^j where j is an integer.

b) Let the range of numbers representable as infinite continued fractions using this digit set be [a,b]. We require that this range form a continuum between a and b.

c) The range [1/2, 1] should be a subset of the range [a, b].

d) The cardinality of the digit set should be as low as possible.

e) It should be possible to develop a selection procedure for our algorithm, with this digit set.

The set {1, 2} does not satisfy the requirement (b). The set {1, 1/2} satisfies all requirements except (e). The reason for this is that, with this set, every number representable as an infinite continued fraction, has a unique representation. We will see later in this paper, that a certain amount of redundancy in representation is necessary to satisfy the requirement (e). The set {1, 1/2, 1/4} satisfies all five requirements; so now we focus our attention on this digit set.

The requirement (a) is clearly satisfied. It is easily shown that the range [a, b], is approximately [0.39, 1.56] with this digit set. Thus the requirement (c) is satisfied. The requirement (d) is also satisfied. To show that the requirement (b) is satisfied we can proceed as follows.

First any number $f_1 \epsilon [a, b]$ can be expanded as a continued fraction as follows. Let

$$f_1 = \frac{1}{q_1 + f_2} \quad \text{and in general, let}$$

$$f_i = \frac{1}{q_i + f_{i+1}} .$$

If $a \leq f_i < 1/2$ then choose $q_i = 1$.

If $1/2 \leq f_i < 1$ then choose $q_i = 1/2$.

If $1 \leq f_i \leq b$ then choose $q_i = 1/4$.

It can easily be verified that with $f_1 \epsilon [a, b]$ and using the above rules, $f_i \epsilon [a, b]$ for all i. Therefore the above rules can be used for all $i \geq 1$. We call such a method of expansion a consistent method of expansion. By an expansion of f_1 to k terms is meant the fraction $\frac{1}{q_1} \, + \, \frac{1}{q_2} \, + \, \cdots \, + \, \frac{1}{q_k}$. Next we use the following theorem, which we state without proof.

<u>Theorem 1</u>: For a number $f_1 \epsilon [a, b]$, if there is a consistent method of expansion of f_1 in the form of a continued fraction, then such an expansion converges to the value f_1 as the number of terms in the expansion increases, provided that the smallest element in the digit set is greater than 0 [5].

Thus every number in [a, b] has an infinite continued fraction expansion with the digit set {1, 1/2, 1/4} and hence the requirement (b) is satisfied.

We devote the rest of this section to show that the requirement (e) is satisfied.

We restrict the problem to $b_k \geq 0$.

Let $f_i = \dfrac{C_{k-i}}{b_{k-i} + u}$ be expanded to $f_i = \dfrac{1}{q_{i+1} + f_{i+1}}$.

Given that $0.39 \leq f_i \leq 1.56$, we have to find $q_{i+1} \in \{1, 1/2, 1/4\}$ such that $0.39 \leq f_{i+1} \leq 1.56$. From these, we get,

for $0.39\ (b_{k-i}+u) \leq c_{k-i} \leq 0.72\ (b_{k-i}+u)$; choose $q_{i+1} = 1$.

for $0.485\ (b_{k-i}+u) \leq c_{k-i} \leq 1.124\ (b_{k-i}+u)$; choose $q_{i+1} = 1/2$.

and

for $0.553\ (b_{k-i}+u) \leq c_{k-i} \leq 1.56\ (b_{k-i}+u)$; choose $q_{i+1} = 1/4$.

The regions where two choices are allowed are,

$0.485\ (b_{k-i}+u) \leq c_{k-i} \leq 0.72\ (b_{k-i}+u)$ then $q_{i+1} = 1/2$ or 1,

and

$0.553\ (b_{k-i}+u) \leq c_{k-i} \leq 1.124\ (b_{k-i}+u)$ then $q_{i+1} = 1/4$ or 1/2.

Both these are triangular wedges in the (c_{k-i}, b_{k-i}) plane. We will call these the (1/2 & 1) and the (1/4 & 1/2) overlap regions, respectively. Clearly these wedges vary with u. To get a selection line which decides between $q_{i+1} = 1/2$ or 1 and which is u-independent, (since u is unknown) we should first take the intersection of all (1/2 & 1) regions as u varies over the range [1/2, 1] and then take a line which is completely within this intersection. A similar statement can be made about the (1/4 & 1/2) region but unfortunately the resulting triangular wedges are not yet wide enough for our problem. It is clear that if we let u vary over a smaller range, we shall have wider overlap regions. Thus partitioning the u-range into three subranges, namely, $I_1 = [1/2, 5/8)$, $I_2 = [5/8, 3/4)$ and $I_3 = [3/4, 1]$ works well. It is clear that from the given values of c_k and b_k it is simple to determine the subrange for root u with shift, add and comparison operations

only. For example,

$$c_k - 1/2\ b_k \geq 1/4 \text{ and } c_k - 5/8\ b_k < 25/64 \Rightarrow u \in I_1.$$

Now we ask for three selection procedures for these three subranges of u. First we discuss the case of subrange $I_1 = [1/2, 5/8)$. The (1/2 & 1) overlap region is given by,

$$0.485\ (b_{k-i}+5/8) \leq c_{k-i} \leq 0.72\ (b_{k-i}+1/2)$$

Similarly, the (1/4 & 1/2) overlap region is given by,

$$0.553\ (b_{k-i}+5/8) \leq c_{k-i} \leq 1.12\ (b_{k-i}+1/2).$$

We show these regions on the (c_{k-i}, b_{k-i}) plane, in figure 5.1. The upper and the lower bounds of the (1/4 & 1/2) region are labelled A and B respectively, and those for the (1/2 & 1) region are labelled C and D. We also show the greatest upper bound $c_{k-i} = 1.56\ (b_{k-i}+5/8)$ as line H and the least lower bound $c_{k-i} = 0.39\ (b_{k-i}+1/2)$ as line L. We also draw two selection lines S1 and S2, which are $c_{k-i} = b_{k-i} + 1/2$ and $c_{k-i} = 1/2\ b_{k-i} + 5/16$ respectively. Notice that the coefficients in these lines are chosen to be "simple" binary numbers. For any point in the region enclosed by line H and S1 we choose $q_{i+1} = 1/4$. For any point between S2 and L, we choose $q_{i+1} = 1$ and otherwise we choose $q_{i+1} = 1/2$. Notice that with these rules our choice could be erroneous in certain regions. Where this happens, we call these regions the forbidden regions. The quadrilateral enclosed by lines H, B, S1 and L is the (1/4 & 1/2) forbidden region and the quadrilateral enclosed by lines H, S2, C and L is the (1/2 & 1) forbidden region. We have to make sure that for no value of i, the point (c_{k-i}, b_{k-i}) lies in one of these regions. This

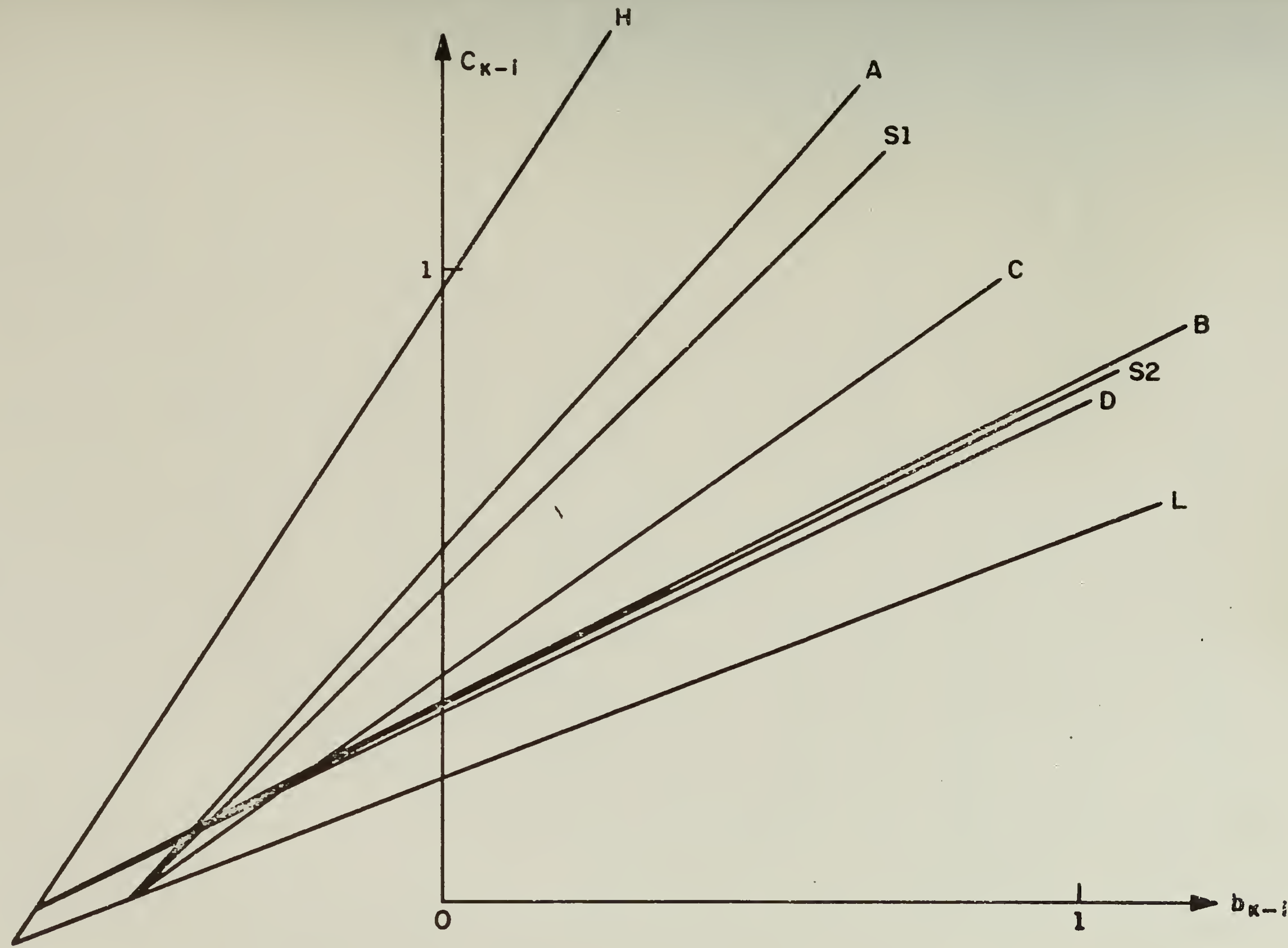

Figure 5.1

we do in appendix I in the proof of convergence. A similar treatment can be given to the other two subranges I_2 and I_3. For the subrange I_2, the selection lines S1 and S2 are, $c_{k-i} = b_{k-i} + 5/8$ and $c_{k-i} = 1/2\ b_{k-i} + 3/8$ respectively. For the subrange I_3, the selection lines S1 and S2 are $c_{k-i} = b_{k-i} + 3/4$ and $c_{k-i} = 1/2\ b_{k-i} + 1/2$ respectively.

Although this general selection procedure is valid for all $i \geq 0$, we want to use a special procedure for $i = 0$ so that when we make tests for the subrange determination, we also find q_1 on the basis of the same tests.

For $i = 0$, $f_i = u$. Then from our previous analysis, we have,

$0.485 \leq u \leq 1.124$ then $q_1 = 1/2$

and

$0.39 \leq u \leq 0.72$ then $q_1 = 1$.

Thus we can choose $q_1 = 1$ for all $u \in I_1$ and $q_1 = 1/2$ for all $u \in I_2$ or I_3.

We now give the complete algorithm A.

A_0: [Check]

If $b_k < o$ then exit, no solution; otherwise

If $(c_k - 1/2 b_k) < 1/4$ or If $(c_k - b_k > 1)$

then exit, no solution;

A_1: [Subrange]

If $c_k - 5/8\ b_k < 25/64$ then set $q_1 \leftarrow 1$,

$K1 \leftarrow 1/2$, $K2 \leftarrow 5/16$ and go to

step A_2; otherwise set $q_1 \leftarrow 1/2$;

If $c_k - 3/4\ b_k < 9/16$ then set

$K1 \leftarrow 5/8$, $K2 \leftarrow 3/8$ and go to step A_2;

otherwise set

$K1 \leftarrow 3/4$, $K2 \leftarrow 1/2$;

A_2: [Initialize]

Set $P_o \leftarrow 0$, $Q_o \leftarrow P_1 \leftarrow 1$, $Q_1 \leftarrow q_1$;

Set $b_{k-1} \leftarrow q_1 c_k$, $c_{k-1} \leftarrow 1 + q_1 (b_k - b_{k-1})$;

Set $i \leftarrow 2$;

A_3: [Selection]

If $c_{k-i+1} > (b_{k-i+1} + K1)$ then

set $q_i \leftarrow 1/4$ and go to step A_4; otherwise

If $c_{k-i+1} \leq 1/2\ b_{k-i+1} + K2$ then

set $q_i \leftarrow 1$ and go to step A_4; otherwise

set $q_i \leftarrow 1/2$;

A_4: [Advance] Set

$b_{k-i} \leftarrow q_i c_{k-i+1} - q_{i-1} c_{k-i+2} + b_{k-i+2}$,

$c_{k-i} \leftarrow q_i (b_{k-i+1} - b_{k-i}) + c_{k-i+2}$,

$P_i \quad \leftarrow q_i P_{i-1} + P_{i-2}$

$Q_i \quad \leftarrow q_i Q_{i-1} + Q_{i-2}$

$i \quad \leftarrow i + 1$;

A_5: [Loop Test] If $i \leq i_{max}$ then go to step A_3;

A_6: [Final] $u\ (=ROOT_1) \leftarrow P_i/Q_i$,

$v \leftarrow b_k + u$;

Note: The value of i_{max} will be decided by the machine precision, in case, this algorithm is implemented in hardware. If this algorithm is implemented in software, however, the value of i_{max} will be decided by the allowable error.

6. Recent Related Work

In the preceding sections, the discovery of the first continued fraction algorithm and its method of application have been described. In this regard, the exposition is historically ordered. The purpose of this section is to describe briefly the results of more recent research.

A study of the derivation of the quadratic algorithm of equations 3.2 has indicated that the requirement that $p_i = 1$ for all i is unnecessary [3]. Equations 3.2 then become

$$
\begin{aligned}
b_{k-n} &= \frac{q_n}{p_n} c_{k-n+1} - \frac{q_{n-1}}{p_{n-1}} c_{k-n+2} + b_{k-n+2} \\
c_{k-n} &= - q_n b_{k-n} + q_n b_{k-n+1} + \frac{p_n}{p_{n-1}} c_{k-n+2}
\end{aligned}
\qquad 6.1
$$

Selection rules for the digit sets $p_i \in \{1/2, 1\}$ and $q_i \in \{1/2, 1\}$ have been determined.

In a companion paper[4], it is shown that the Ricatti equation

$$
y' + ay^2 + by + c = 0 \qquad 6.2
$$

leads to relatively simple recursions if the partial numerators p_i and partial denominators q_i of the associated continued fraction are simple in the binary sense. Since both tan x and e^x satisfy the Ricatti equation for particular choices of a, b, and c, there is some hope that useful continued fraction algorithms for these functions can be found. Attempts to find selection rules for tan x have thus far been unsuccessful, and have not been attempted for the exponential.

The derivation of the recursion relations for the Ricatti equation suggested a similar derivation for the quadratic equation, and led to a second set of recursion relations for the quadratic. The special selection

procedure for $i = 0$ described in section 5 suffices for selection rules for this second quadratic algorithm. Recursion relations for higher order polynomials can also be found by this method.

7. Conclusions

It should be emphasized that the primary purpose of this paper is to point out that hardware construction can be based on representations of numbers other than positional notation. It seems quite clear that the use of continued products yields a useful set of algorithms which can share the same hardware in a feasible and practical manner using current technology.

The discussion of continued fractions presented here is a case study of the problems which arise when a different representation of numbers is proposed. The research on the use of continued fractions is incomplete; the results obtained thus far do not justify hardware construction based on continued fractions.

It seems appropriate, therefore, to conclude with a list of questions for future research. These include:

1) Can the set of algorithms soluble with continued products be extended?
2) How can the set of algorithms based on the use of continued fractions be extended? Can feasible selection rules for each algorithm be found?
3) What additional representations of numbers exist? What is their potential usefulness?

APPENDIX I

The Proof of Convergence of Algorithm A

To prove convergence, we only need to show that the selection procedure in step A_3 of Algorithm A is a consistent method, and then using theorem 1, we have the required result.

To show consistency, first notice that,

$$f_i = \frac{c_{k-i}}{b_{k-i} + u} \quad \text{and} \quad a \leq f_i \leq b$$

for all i. Thus a $(b_{k-i}+u) \leq c_{k-i} \leq b\ (b_{k-i}+u)$ is always satisfied, and we are well within the bounds H and L of figure 5.1. We also need to show that for any value of $i \geq 1$, the point (c_{k-i}, b_{k-i}) does not fall in any one of the forbidden regions. It is necessary to treat each of the ranges I_1, I_2 and I_3 separately. We only treat the range I_1 here, others being similar. We use the following result [5].

For all $i > o$; c_{k-i} and b_{k-i} satisfy one equation to the line $c_{k-i} = \frac{P_i}{Q_{i-1}} - \frac{Q_i}{Q_{i-1}}\ (b_{k-i}-b_k)$. The line of closest approach to the forbidden region or alternatively, the left most line is given by

$$c_{k-i} = \left(\frac{P_i}{Q_{i-1}}\right)_{min} - \left(\frac{Q_i}{Q_{i-1}}\right)_{min} b_{k-i}$$

since $b_k \geq o$. We wish to show that this line is within the allowed region in figure 5.1.

We have, $1/4 \leq c_k < 25/64$ and $b_k = o$ then with $q_1 = 1$, the above line for $i = 1$ is $c_{k-1} = 1 - b_{k-1}$, $1/4 \leq b_{k-1} < 25/64$. This is shown (labelled P) in figure A.1. Next consider $i \geq 3$ and i odd. We first get a lower bound on $\frac{P_i}{Q_i}$. We use a theorem [6], which states that odd ordered convergents approach the value of an infinite continued fraction from above (and even ordered convergents approach from below).

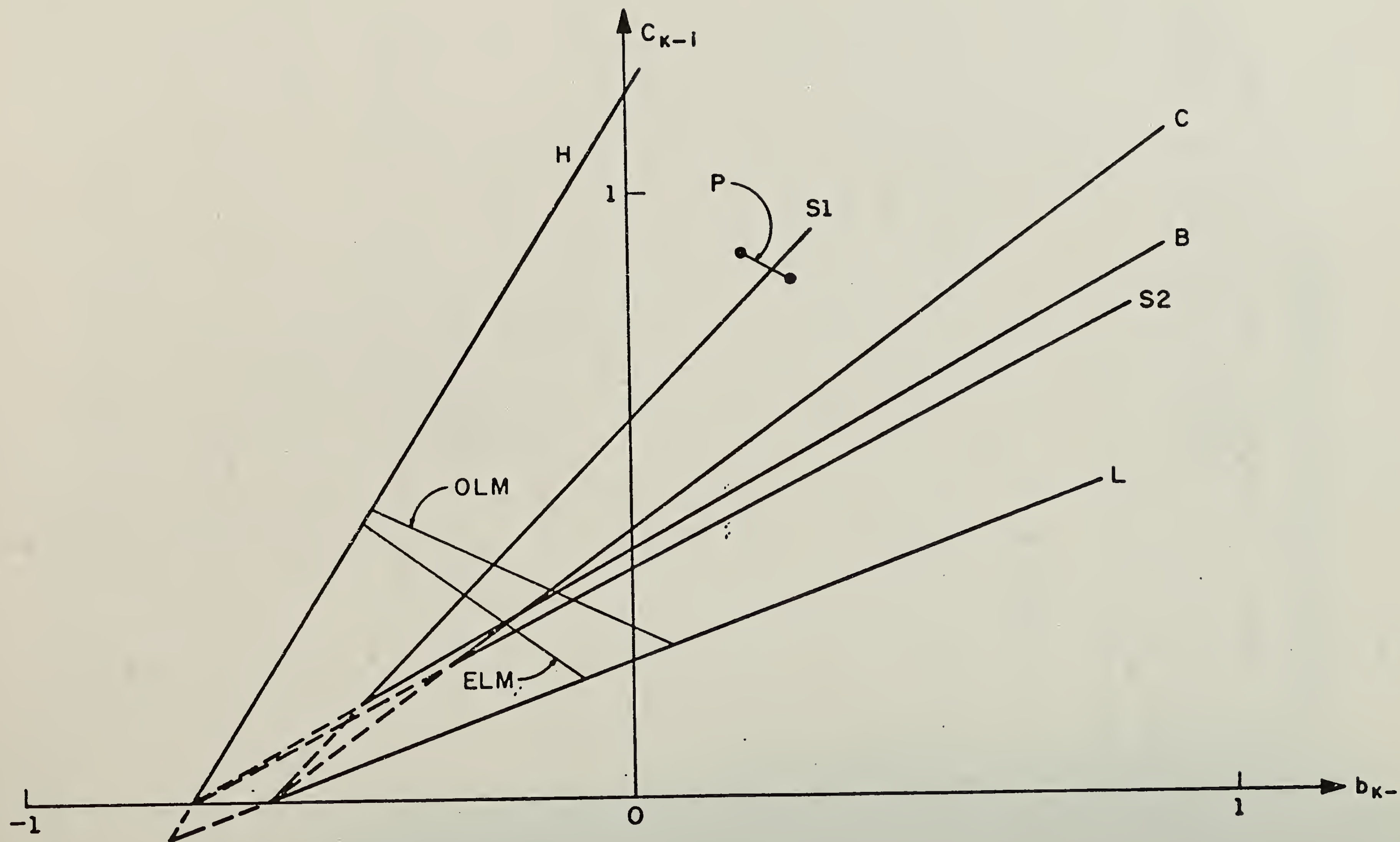

Figure A.1

Thus $\left(\frac{P_i}{Q_i}\right)_{i \text{ odd}} \geq \min_{u \in I_1} (u) = 1/2.$

Now
$$\frac{Q_i}{Q_{i-1}} = q_i + \cfrac{1}{q_{i-1} + \cfrac{1}{q_{i-2} + \cdots \frac{1}{q_1}}}$$

$$\left(\frac{Q_i}{Q_{i-1}}\right)_{\substack{i \geq 3 \\ \text{odd}}} \geq 1/4 + \cfrac{1}{1 + \cfrac{1}{1/4}} = 9/20$$

Thus $\left(\frac{P_i}{Q_{i-1}}\right)_{\substack{i \geq 3 \\ \text{odd}}} \geq 9/40.$

Thus the left most line (labelled OLM in figure A.1) for odd values of i, is given by

$$c_{k-i} = 9/40 - 9/20\ b_{k-i}.$$

Next consider the case of even values of $i \geq 2$. It can be seen (from figure A.1) that $q_2 = 1/2$ for all $u \in I_1$. Then $\frac{P_2}{Q_2} = 1/3$. Now again using the stated theorem on convergents, $\left(\frac{P_i}{Q_i}\right)_{\substack{i \geq 2 \\ \text{even}}} \geq 1/3.$

Next observe that,

$$\left(\frac{Q_i}{Q_{i-1}}\right)_{i \text{ even}} \geq \frac{1}{4} + \frac{1}{1} + \frac{1}{\frac{1}{4}} + \frac{1}{1} + \frac{1}{\frac{1}{4}} + \quad \cdots\cdots\cdots \quad + \frac{1}{1} \text{ finite}$$

$$> \frac{1}{4} + \frac{1}{1} + \frac{1}{\frac{1}{4}} + \frac{1}{1} + \cdots\cdots\cdots\cdots \text{ infinite.}$$

$$\simeq 0.640388$$

Thus $\left(\frac{P_i}{Q_i}\right)_{\substack{i \geq 2 \\ \text{even}}} \geq 0.2135.$

Thus the left most line (labelled ELM in figure A.1) is given by, $C_{k-i} = 0.2135 - 0.64\ b_{k-i}$. It is clear from figure A.1 that we never transcend into the forbidden regions. Thus we have satisfied the consistency requirement

As a matter of practical interest, we mention here that c_{k-i} and b_{k-i} remain bounded in algorithm A. In particular bounds on c_{k-i} are given by $0 \leq C_{k-i} \leq 3.57 + 1.2\ b_k$.

Bibliography

[1] DeLugish, Bruce G., A Class of Algorithms for Automatic Evaluation of Certain Elementary Functions in a Binary Computer, Ph.D. thesis, University of Illinois, June 1950; also Department of Computer Science Report No. 399.

[2] Ercegovac, Milos, Radix 16 Evaluation of Some Elementary Functions, M.S. thesis in preparation at the Department of Computer Science, University of Illinois, Urbana.

[3] Bracha, Amnon, private communication.

[4] Bracha, Amnon, Application of Continued Fractions for Fast Evaluation of Certain Functions on a Digital Computer.

[5] Trivedi, K. S., An Algorithm for the Solution of a Quadratic Equation using Continued Fractions, M.S. thesis in preparation at the Department of Computer Science, University of Illinois, Urbana.

[6] Wall, H., Analytic Theory of Continued Fractions, Van Nostrand, New York, 1950.

BIBLIOGRAPHIC DATA SHEET	1. Report No. UIUCDCS-R-72-542	2.	3. Recipient's Accession No.

4. Title and Subtitle The Status of Investigations into the Use of Continued Fractions for Computer Hardware	5. Report Date August 1972 6. Date of issue
7. Author(s) James E. Robertson and Kishor Trivedi	8. Performing Organization Rept. No.
9. Performing Organization Name and Address Department of Computer Science University of Illinois Urbana, Illinois 61801	10. Project/Task/Work Unit No. 11. Contract/Grant No. NSF GJ 813
12. Sponsoring Organization Name and Address National Science Foundation Washington, D. C.	13. Type of Report & Period Covered Research 14.

15. Supplementary Notes

16. Abstracts

The purpose of this paper is to demonstrate that representations of numbers other than positional notation may lead to practical hardware realizations for the digital calculation of classes of algorithms. It is the authors' opinion that practicality of the use of continued products has been demonstrated. This paper describes current research in the use of continued fractions. Although practicality has not been demonstrated, theoretical results are promising, and the results thus far are presented as a case study of the difficulties which arise when use of a new representation is attempted.

17. Key Words and Document Analysis. 17a. Descriptors

Continued Products, Continued Fractions, Computer Arithmetic, Hardware, Selection Rules, Radix, Representation of Numbers, Ricatti Equation.

17b. Identifiers/Open-Ended Terms

17c. COSATI Field/Group

18. Availability Statement	19. Security Class (This Report)	21. No. of Pages
Unlimited Release	UNCLASSIFIED	29
	20. Security Class (This Page) UNCLASSIFIED	22. Price

FORM NTIS-35 (10-70) USCOMM-DC 40329-P71